

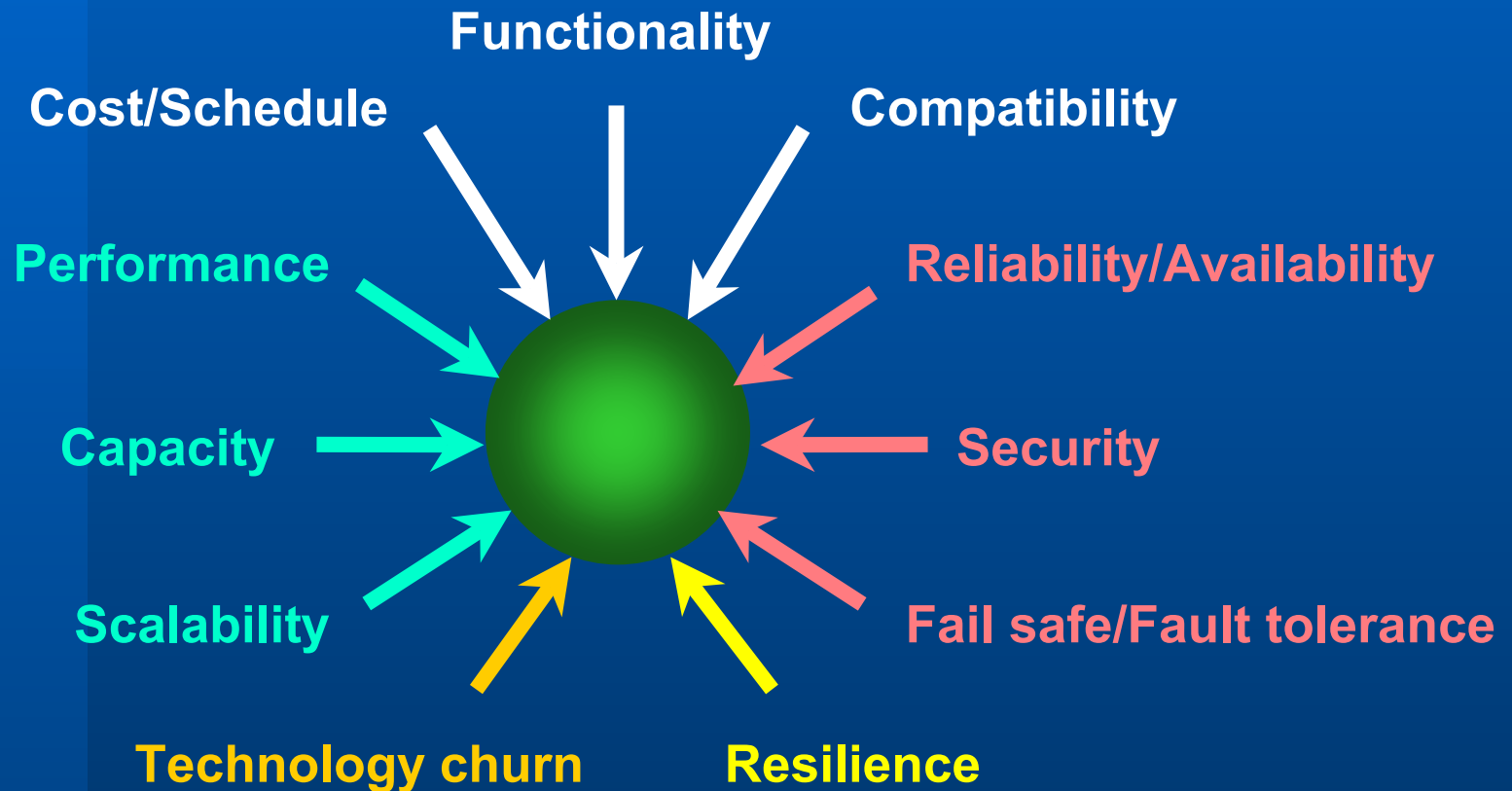
New Paradigms For Large-Scale Software Design And Development

Grady Booch

egb@rational.com

Rational
the e-development company

Forces In Software



Points Of Friction

- Inherent complexity
- Start up
- Work product collaboration
- Communication
 - Knowledge and experience
 - Status
 - Project memory
 - Stakeholder cooperation
- Time starvation
- Stuff that doesn't work

Points Of Friction

- Inherent complexity
- Start up
- Work product
- Communication
 - Knowledge
 - Status
 - Project
 - Stakeholder relationships
- Time starvation
- Stuff that doesn't work

Software development is ultimately a human activity

Creating A Frictionless Surface

- **Inherent complexity**
 - Raise the level of abstraction by composing systems from components and services and by transforming models from reference architectures/mechanisms
- **Start up**
 - Make the virtual project environment just an URL away
- **Work product collaboration**
 - Provide artifact storage with change management, meta-knowledge, and semantically rich tools
- **Communication**
 - Provide discussion mechanisms, virtual meetings, project dashboard, and workflow automation
- **Time starvation**
 - Deliver virtual agents
- **Stuff that doesn't work**

Disruptive Technologies

- Collaborative development environments: using the Web to change the nature of software development
 - Web-based
 - Artifact-centric
 - Individual tools disappear
 - Different views for different stakeholders
- Reconciliation of low/high ceremony processes
- Model-driven development
- Languages for exotic computational models